1. class A {

int i;

int j;

//const.

A( ) {

i = 1;

j = 2;

}

}

class Output {

public static void main(String args[])

{

A obj1 = new A();

A obj2 = new A();

System.out.print(obj1.equals(obj2));//false

}

}

2. class ThisDemo {

int a;

int b;

//const

ThisDemo(int a, int b){ //local vars

// a=a;

//b=b;

}

public static void main(String[] args) {

ThisDemo obj = new ThisDemo(10, 20);

System.out.println(obj.a);//0

System.out.println(obj.b);//0

}

}

1. public class ThisDemo {

int a;//0

int b;//0

ThisDemo(int a, int b){

a=a;

this.b=b;

}

public static void main(String[] args) {

ThisDemo obj = new ThisDemo(10, 20);

System.*out*.println(obj.a);//0

System.*out*.println(obj.b);//20

}

}

4. public class ThisDemo {

int a;//0🡪1

int b;//0->2

ThisDemo(int a, int b){//1,2

this.a=a;//a=1

this.b=b;//b=2

}

public static void main(String[] args) {

ThisDemo obj = new ThisDemo( );//compilation Error

System.out.println(obj.a);//1

System.out.println(obj.b);//2

}

}

5. public class TestClass {

int x;

int y;

String st ;

public TestClass() {

System.*out*.println("Zero Args Const. called");//1

}

public TestClass(int x, int y) {

this();

this.x =x;

this.y =y;

System.*out*.println("Two Args Const. called");//2

}

public static void main(String[] args) {

TestClass tc = new TestClass(1,2);

System.*out*.println("x= "+ tc.x);//0

System.*out*.println("y= "+ tc.y);//0

}

}

6. public class TestClass {

int a,b;

public TestClass(int a, int b) {

this.a=a;

this.b=b;

}

Static void show(int x) {

System.*out*.println("method");//1

}

//non static

void print() {

show();

System.*out*.println(a);//1

System.*out*.println(b);//2

}

public static void main(String[] args) {

TestClass tc = new TestClass(1,2);//

tc.print();

}

}

7. Can we call non static method in a constructor using ‘this’ keyword?

8. public class ThisDemo {

int a;

int b;

ThisDemo(int x, int y){

this= new ThisDemo();///

System.*out*.println("Two argument constructor called.");

this.x=x;

this.y=y;

}

public static void main(String[] args) {

ThisDemo obj = new ThisDemo(10, 20);

System.*out*.println(obj.a);

System.*out*.println(obj.b);

}

}

10. public class B

{

int a;//

public int getA() {

return a;

}

public void setA(int a) {

this.a = a;

}

B show(){

return this;

}

public static void main(String[] args) {

B obj = new B();

obj.setA(10);

System.out.println(obj.getA());

B obj2= obj.show();

System.out.println(obj2.getA());

}

}

11. What is the output of this program?

class Test{

public void show(){

this=null; //L.H. S must be variable

}

public static void main(String args[])

{

System.out.println("Instance class");

}

}

12. What is the output of this program?

class access{

static int x;

void increment(){

x++;

}

}

class static\_use {

public static void main(String args[])

{

access obj1 = new access();

access obj2 = new access();

obj1.x = 0;

obj1.increment();

obj2.increment();

System.out.println(obj1.x + " " + obj2.x);

} }

13. What is the output of this program?

class static\_out {

static int x;

static int y;

void add(int a , int b){

x = a + b;

y = x + b;

}

}

class static\_use {

public static void main(String args[])

{

static\_out obj1 = new static\_out();

static\_out obj2 = new static\_out();

int a = 2;

obj1.add(a, a + 1);

obj2.add(5, a);

System.out.println(obj1.x + " " + obj2.y);

}

}

14.A class called **circle** is designed. It contains:

* Two instance variables: radius (of the type double) and color (of the type String), with default value of 1.0 and "red", respectively.
* Two *overloaded* constructors - a *default* constructor with no argument, and a constructor which takes a double argument for radius.
* Two public methods: getRadius() and getArea(), which return the radius and area of this instance, respectively.

15. Write a program to use of const. overloading and this keyword.